

03500.017681



PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	
	:	
ATSUSHI DATE)	
	:	Group Art Unit: 2182
Application No.: 10/670,302)	
	:	
Filed: September 26, 2003)	
	:	
For: SYSTEM CONTROLLER)	
USING PLURAL CPU's	:	
	:	
)	February 23, 2004

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUBMISSION OF PRIORITY DOCUMENT

Sir:

In support of Applicant's claim for priority under 35 U.S.C. § 119, enclosed is
a certified copy of the following foreign application:

JP 2002-286050, filed September 30, 2002.

Applicant's undersigned attorney may be reached in our Costa Mesa, California office by telephone at (714) 540-8700. All correspondence should continue to be directed to our address given below.

Respectfully submitted,


Attorney for Applicant

Registration No. 32622

FITZPATRICK, CELLA, HARPER & SCINTO
30 Rockefeller Plaza
New York, New York 10112-3800
Facsimile: (212) 218-2200

CA_MAIN 77322v1

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: 2002年 9月30日

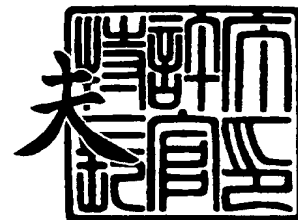
出願番号
Application Number: 特願2002-286050
[ST. 10/C]: [JP2002-286050]

出願人
Applicant(s): キヤノン株式会社

2003年10月21日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫





【書類名】 特許願

【整理番号】 4211083

【提出日】 平成14年 9月30日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 G06F 13/38

【発明の名称】 システムコントローラ

【請求項の数】 2

【発明者】

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社
内

【氏名】 伊達 厚

【特許出願人】

【識別番号】 000001007

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号

【氏名又は名称】 キャノン株式会社

【代表者】 御手洗 富士夫

【電話番号】 03-3758-2111

【代理人】

【識別番号】 100090538

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会社
内

【弁理士】

【氏名又は名称】 西山 恵三

【電話番号】 03-3758-2111

【選任した代理人】

【識別番号】 100096965

【住所又は居所】 東京都大田区下丸子 3 丁目 3 0 番 2 号キャノン株式会
社内

【弁理士】

【氏名又は名称】 内尾 裕一

【電話番号】 03-3758-2111

【手数料の表示】

【予納台帳番号】 011224

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9908388

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 システムコントローラ

【特許請求の範囲】

【請求項 1】 共有バスを介して接続される複数の CPU と複数のメモリまたは I/O デバイスとを、リード命令とリードデータリターンとが分離可能なバスを介して接続されたシステムコントローラであって、

命令を発行した CPU と、その命令の転送先と、転送保留中の CPU とを識別する識別情報を保持する保持手段と、

リード時に、前記保持手段の保持内容に基づいて、リターンデータが転送命令の発行順と入れ替わらないように制御する順序制御手段と、

一旦シリアルライズされて前記共有バスを介して行なわれた転送を、複数の接続経路を用いて並行に発行する発行手段とを備えたことを特徴とするシステムコントローラ。

【請求項 2】 前記共有バスに出力されたトランザクション要求を、当該共有バス上でバススヌーピング後にキューイングする手段を更に有することを特徴とする請求項 1 に記載のシステムコントローラ。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、複数 CPU を用い、クロスバースイッチを用いたバス接続を有するシステムコントローラで、特に、スキャニング、プリンティングデバイス、ネットワークインターフェース等の制御を目的とする複合機器のコントローラに関するものである。

【0 0 0 2】

【従来の技術】

①従来、図 1 2 に示すような、複数の CPU 2 0 0 1、メモリコントローラ、DMAC 等を共通のバス 2 1 2 6 に接続したマルチプロセッサシステムが存在する。

【0 0 0 3】

②また、複数のCPUを、同時接続可能な複数のバススイッチのマスタポートに接続した例も存在する。

【0004】

③更に、共通バスをバススイッチのマスタポートのひとつに接続する構成も提案がされている。

【0005】

【発明が解決しようとする課題】

しかしながら、上記従来技術には、以下のような問題があった。

【0006】

上記①の従来例では、バススヌーピングによるキャッシュメモリのコヒーレンシ管理、アトミックトランザクションの実現等が簡易になる反面、共通バスに接続されたマスタデバイスが一度に一つしか使用できないという問題点、共有バスに接続されるデバイス数が多くなり、動作周波数が制限されるという問題点等により、高性能化が難しいという問題があった。

【0007】

上記②の従来例では、同時接続性、バス負荷の低減による高性能化は可能な反面、一方のCPUより他方のCPUのバストランザクションが観測できず、キャッシュメモリのコヒーレンシ管理、ロードリンク、ストアコンディショナルによるアトミックトランザクションのサポートが困難であるという問題があった。そこで、この問題を解決するために、一方のトランザクションをターゲットスレーブに伝達する前に他のCPUに対して転送し、スヌーピングを行わせることにより、キャッシュのコヒーレンシ保持及びアトミックトランザクションを実現する方法が提案されている。

【0008】

しかし、この方法では、ライト動作がスヌーピング終了まで保留され、CPUの性能を制限してしまう。またライトバッファを実装する場合に、一方のライト動作終了前に他方のCPUがライト動作を検出できず、リンクビットをブレイク出来ないため、アトミックトランザクションが保証できないという問題があった。

【0009】

上記③の方式は、上述の問題を解決するが、共通バスに接続された一方のCPUがアクセス時間の長い低速のデバイスに転送要求を発行すると、もう一方のCPUがメモリ等の高速デバイスにアクセスしようとした場合でも、最初のCPUの転送終了までアクセスが保留されてしまい、複数CPUを用いた場合の性能の向上を制限してしまうという問題があった。

【0010】**【課題を解決するための手段】**

上記課題を解決するために、本発明によれば、共有バスを介して接続される複数のCPUと複数のメモリまたはIOデバイスとを、リード命令とリードデータリターンとが分離可能なバスを介して接続されたシステムコントローラに、命令を発行したCPUと、その命令の転送先と、転送保留中のCPUとを識別する識別情報を保持する保持手段と、リード時に、前記保持手段の保持内容に基づいて、リターンデータが転送命令の発行順と入れ替わらないように制御する順序制御手段と、一旦シリアルライズされて前記共有バスを介して行なわれた転送を、複数の接続経路を用いて並行に発行する発行手段とを備える。

【0011】**【発明の実施の形態】****〔第一の実施形態〕**

以下に、本発明の装置及びその動作について詳細に説明する。

【0012】**<全体構成>**

全体構成図を図1に示した。

【0013】

Controller Unit (2000) は画像入力デバイスである Scanner (2070) や画像出力デバイスである Printer (2095) と接続し、一方では LAN (2011) や公衆回線 (WAN) <2051> と接続することで、画像情報やデバイス情報の入出力、PDLデータのイメージ展開を行う為のコントローラである。

【0014】

CPU (2001) はシステム全体を制御するプロセッサである。本実施形態では2つのCPUを用いた例を示した。これら二つのCPUは、共通のCPUバス<2126>に接続され、さらに、システムバスブリッジ (2007) に接続される。

【0015】

システムバスブリッジ (2007) は、バススイッチであり、CPUバス (2126)、RAMコントローラ (2124)、ROMコントローラ (2125)、IOバス1 (2127)、サブバススイッチ (2128)、IOバス2 (2129)、画像リングインターフェース1 (2147)、画像リングインターフェース2 (2148) が接続される。

【0016】

サブバススイッチ (2128) は、第2のバススイッチであり、画像DMA1 (2130)、画像DMA2 (2132)、フォント伸張部 (3134)、ソート回路 (2135)、ビットマップトレース部 (2136) が接続され、これらのDMAから出力されるメモリアクセス要求を調停し、システムバスブリッジへの接続を行う。

【0017】

RAM (2002) はCPU (2001) が動作するためのシステムワークメモリであり、画像データを一時記憶するための画像メモリでもある。RAMコントローラ (2124) により制御される、本実施形態では、ダイレクトRDRA Mを採用する例を示した。

【0018】

ROM (2003) はブートROMであり、システムのブートプログラムが格納されている。ROMコントローラ (2125) により制御される。

【0019】

画像DMA1 (2130) は、画像圧縮部 (3131) に接続し、レジスタアクセスリング (2137) を介して設定された情報に基づき、画像圧縮部 (2131) を制御し、RAM (2002) 上にある非圧縮データの読み出し、圧縮、

圧縮後データの書き戻しを行う、本実施形態では、J P E Gを圧縮アルゴリズムに採用した例を示した。

【0020】

画像DMA 2 (2132) は、画像伸張部 (2133) に接続し、レジスタアクセスリング (2137) を介して設定された情報に基づき、画像伸張部 (2133) を制御し、RAM (2002) 上にある圧縮データの読み出し、伸張、伸張後データの書き戻しを行う、本実施形態では、J P E Gを伸張アルゴリズムに採用した例を示した。

【0021】

フォント伸張部 (2134) は、LANインターフェース (2010) 等を介して外部より転送されるPDLデータに含まれるフォントコードに基づき、ROM (2003) もしくは、RAM (2002) 内に格納された、圧縮フォントデータの伸張を行う。本実施形態では、F B Eアルゴリズムを採用した例を示した。

【0022】

ソート回路 (2135) は、PDLデータを展開する段階で生成されるディスプレイリストのオブジェクトの順番を並び替える回路である。ビットマップトレース回路 (2136) は、ビットマップデータより、エッジ情報を抽出する回路である。

【0023】

I Oバス1 (2127) は、内部I Oバス的一种であり、標準バスであるUSBバスのコントローラ、USBインターフェース (2138)、汎用シリアルポート (2139)、インタラプトコントローラ (2140)、G P I Oインターフェース (2141) が接続される。I Oバス1には、バスアービタ (図示せず) が含まれる。

【0024】

操作部 I / F (2006) は操作部 (U I) (2012) とインターフェース部であり、操作部 (2012) に表示すべき画像データを操作部 (2012) に対して出力する。また、操作部 (2012) から本システム使用者が入力した情

報を、CPU (2001) に伝える役割をする。

【0025】

IOバス2 (2129) は内部IOバスの一種であり、汎用バスインターフェース1及び2 (2142) と、LANコントローラ (2010) が接続される。IOバス2にはバスアービタ (図示せず) が含まれる。

【0026】

汎用バスインターフェース (2142) は、2つの同一のバスインターフェースから成り、標準IOバスをサポートするバスブリッジである。本実施形態では、PCIバス (2143) を採用した例を示した。

【0027】

HDD (2004) はハードディスクドライブで、システムソフトウェア、画像データを格納する。ディスクコントローラ (2144) を介して一方のPCIバス (2143) に接続される。

【0028】

LANコントローラ (2010) は、MAC回路 (2145)、PHY/PM D回路 (2146) を介してLAN (2011) に接続し、情報の入出力を行う。

【0029】

Modem (2050) は公衆回線 (2051) に接続し、情報の入出力を行う。

【0030】

画像リングインターフェース1 (2147) 及び画像リングインターフェース2 (2148) は、システムバスブリッジ (2007) と画像データを高速で転送する画像リング (2008) を接続し、タイル化後に圧縮されたデータをRAM (2002) とタイル画像処理部 (2149) との間で転送するDMAコントローラである。

【0031】

画像リング (2008) は、一対の単方向接続経路の組み合わせにより構成される (画像リング1及び画像リング2)。画像リング (2008) は、タイル画

像処理部(2149)内で、画像リングインターフェース3(2101)及びタイル画像インターフェース4(2102)を介し、タイル伸張部(2103)、コマンド処理部(2104)、ステータス処理部(2105)、タイル圧縮部(2106)に接続される。本実施形態では、タイル伸張部(2103)を2組、タイル圧縮部を3組実装する例を示した。

【0032】

タイル伸張部(2103)は、画像リングインターフェースへの接続に加え、タイルバス(2107)に接続され、画像リングより入力された圧縮後の画像データを伸張し、タイルバス(2107)へ転送するバスブリッジである。本実施形態では、多値データにはJ P E G、2値データにはパックピッツを伸張アルゴリズムとして採用した例を示した。

【0033】

タイル圧縮部(2106)は、画像リングインターフェースへの接続に加え、タイルバス(2107)に接続され、タイルバスより入力された圧縮前の画像データを圧縮し、画像リング(2008)へ転送するバスブリッジである。本実施形態では、多値データにはJ P E G、2値データにはパックピッツを圧縮アルゴリズムとして採用した例を示した。

【0034】

コマンド処理部(2104)は、画像リングインターフェースへの接続に加え、レジスタ設定バス(2109)に接続され、画像リングを介して入力したC P U(2001)より発行されたレジスタ設定要求を、レジスタ設定バス(2109)に接続される該当ブロックへ書き込む。また、C P U(2001)より発行されたレジスタ読み出し要求に基づき、レジスタ設定バスを介して該当レジスタより情報を読み出し、画像リングインターフェース4(2102)に転送する。

【0035】

ステータス処理部(2105)は、各画像処理部の情報を監視し、C P U(2001)に対してインタラプトを発行するためのインタラプトパケットを生成し、画像リングインターフェース4に出力する。

【0036】



タイルバス（2107）には、上記ブロックに加え、以下の機能ブロックが接続される。レンダリング部インターフェース（2110）、画像入力インターフェース（2112）、画像出力インターフェース（2113）、多値化部（2119）、2値化部（2118）、色空間変換部（2117）、画像回転部（2030）、解像度変換部（2116）。

【0037】

レンダリング部インターフェース（2110）は、後述するレンダリング部により生成されたビットマップイメージを入力するインターフェースである。レンダリング部とレンダリング部インターフェースは、一般的なビデオ信号（2111）にて接続される。レンダリング部インターフェースは、タイルバス（2107）に加え、メモリバス（2108）、レジスタ設定バス（2109）への接続を有し、入力された、ラスター画像をレジスタ設定バスを介して設定された、所定の方法によりタイル画像への構造変換をすると同時にクロックの同期化を行い、タイルバス（2107）に対し出力を行う。

【0038】

画像入力インターフェース（2112）は、後述するスキャナー用画像処理部（2114）により補正画像処理されたラスターイメージデータを入力とし、レジスタ設定バスを介して設定された、所定の方法によりタイル画像への構造変換とクロックの同期化を行い、タイルバス（2107）に対し出力を行う。

【0039】

画像出力インターフェースは、タイルバスからのタイル画像データを入力とし、ラスター画像への構造変換及び、クロックレートの変更を行い、ラスター画像をプリンタ用画像処理部（2115）へ出力する。

【0040】

画像回転部（2030）は画像データの回転を行う。解像度変換部（2116）は画像の解像度の変更を行う。色空間変換部（2117）はカラー及びグレースケール画像の色空間の変換を行う。2値化部（2118）は多値（カラー、グレースケール）画像を2値化する。多値化部（2119）は2値画像を多値データへ変換する。

【 0 0 4 1 】

外部バスインターフェース部（2 1 2 0）は、画像リングインターフェース 1、2、3、4、コマンド処理部、レジスタ設定バスを介し、CPU（2 0 0 1）により発行された、書き込み、読み出し要求を外部バス 3（2 1 2 1）に変換出力するバスブリッジである。外部バス 3（2 1 2 1）は本実施形態では、プリンター用画像処理部（2 1 1 5）、スキャナー用画像処理部（2 1 1 4）に接続されている。

【 0 0 4 2 】

メモリ制御部（2 1 2 2）は、メモリバス（2 1 0 8）に接続され、各画像処理部の要求に従い、あらかじめ設定されたアドレス分割により、画像メモリ 1 及び画像メモリ 2（2 1 2 3）に対して、画像データの書き込み、読み出し、必要に応じてリフレッシュ等の動作を行う。本実施形態では、画像メモリに S D R A Mを用いた例を示した。

【 0 0 4 3 】

スキャナー用画像処理部（2 1 1 4）では、画像入力デバイスであるスキャナ（2 0 7 0）によりスキャンされた画像データを補正画像処理する。

【 0 0 4 4 】

プリンタ用画像処理部では、プリンタ出力のための補正画像処理を行い、結果を P r i n t e r（2 0 9 5）へ出力する。

【 0 0 4 5 】

レンダリング部（2 0 6 0）は P D Lコードもしくは、中間ディスプレイリストをビットマップイメージに展開する。

【 0 0 4 6 】

〔システム全体〕

本実施形態のネットワークシステム全体の構成図を図 2 に示した。

【 0 0 4 7 】

1 0 0 1 は複合機器で、スキャナとプリンタから構成され、スキャナから読み込んだ画像をローカルエリアネットワーク（以下 LAN）（1 0 1 0）に流したり、LAN から受信した画像をプリンタによりプリントアウトできる。また、ス

キャナから読んだ画像を図示しないFAX送信手段により、PSTNまたはISDN(1030)に送信したり、PSTNまたはISDNから受信した画像をプリンタによりプリントアウトできる。1002は、データベースサーバで、本実施形態の複合機器(1001)により読み込んだ2値画像及び多値画像をデータベースとして管理する。

【0048】

1003は、データベースサーバ(1002)のデータベースクライアントで、データベース(1002)に保存されている画像データを閲覧／検索等できる。

【0049】

1004は、電子メールサーバで、複合機器(1001)により読み取った画像を電子メールの添付として受け取ることができる。1005は、電子メールのクライアントで、電子メールサーバ(1004)の受け取ったメールを受信し閲覧したり、電子メールを送信したり、可能である。

【0050】

1006はHTML文書をLANに提供するWWWサーバであり、複合機器(1001)により、WWWサーバ(1006)で提供されるHTML文書をプリントアウトできる。

【0051】

1007は、ルータでLAN(1010)をインターネット／イントラネット(1012)と連結する。インターネット／イントラネットに、前述したデータベースサーバ(1002)、WWWサーバ(1006)、電子メールサーバ(1004)、複合機器(1001)と同様の装置が、それぞれ1020、1021、1022、1023として連結している。一方、複合機器(1001)は、PSTNまたはISDN(1030)を介して、FAX装置(1031)と送受信可能になっている。

【0052】

また、LAN上にプリンタ(1040)も連結されており、複合機器(1001)により読み取った画像をプリントアウト可能なように構成されている。

【0053】

図3に、本実施形態によるCPUバスインターフェース部のみのブロック図を示した。2つのCPU(2001)が共通バス(2126)に接続される。システムバスブリッジ(2007)内部に実装されるCPUバスインターフェース(10)が、この共通バス(2126)のスレーブとして接続される。CPUバスインターフェース内部には、システムバスブリッジ内部のバス(Ybus)(11)へのマスタポートを2つ持つ。図3では簡潔のため、RAM(2002)、IOバス1(2127)、IOバス2(2129)への接続のみを図示した。

【0054】

図4にシステムバスブリッジ(2007)のブロック図を示す。

【0055】

システムバスブリッジ(2007)は、バススイッチ(3003)によって、複数のバスインターフェースブロックが相互に接続された構成をとる。相互に接続されたバスインターフェースには、MCIF(3001)、BIF(3002)、RCIF(3004)、CPUバスインターフェース(3005)、CIU(3006)、YMIF(3007)、EBIF(3008)、SRIF(3011)、GUIFJI(3012)、GUIF_PO(3013)が含まれる。その他の構成要素として、SRUIF(3010)、REG(3009)が含まれる。

【0056】

MCIF(3001)には、RAMコントローラ(2124)が接続される。BIF(3002)には、IOバス1(2127)が接続される。RCIF(3004)には、ROMコントローラ(2125)が接続される。CIU(3006)は、CPUバス(2126)に接続され、キャッシュヌーピングに必要な情報をCPUバスに対して伝達する。

【0057】

YMIF(3007)には、サブバススイッチ(2128)が接続される。EBIF(3008)には、IOバス2(2129)が接続される。SRIF(3011)には、レジスタアクセスリング(2137)が接続される。GUIF_P



I (3012) には、画像リングインターフェース 2 (2148) が接続される。GUIF_PO (3013) には、画像リングインターフェース 1 (2147) が接続される。SRUIF (3010) は、レジスタアクセスリング (2137) に接続され、システムバスブリッジ内部のプログラミングレジスタ REG (3009) へのデータの読み書きを行う。

【0058】

CPUバスインターフェース (CPUBusIF) (3005) は、本実施形態において最も特徴的な部分であり、スイッチ (3003) 内部のバス (Ybus) と CPUバス (2126) のプロトコル変換を行うバスブリッジである。

【0059】

CPUバスインターフェースの構成をさらに詳しく説明する。図5にCPUバスインターフェースのブロック図を示した。

【0060】

CPUバスインターフェースは、CPUコアの外部バスインターフェースであるCPUバスと、SBB内部バスであるYbusとの間のバスプロトコル変換回路である。CPUバスインターフェース内には、Command_Q (101)、YbusMasterIO (104)、YbusMasterMEM (105)、Decode (102)、Dispatch (103)、ReadReturnArbitor (106)、ReadDataMux (107) が含まれる。

【0061】

CPUバスインターフェースは、CPUバスに定義された図6の転送をサポートする。

【0062】

CPUバスインターフェースのYbusマスタIDは、0000及び0001 (y0__xxx、y1__xxx) である。

【0063】

Command_Q (101) は、CPUバスより発行されたトランザクションコマンドをキューイングするキューである。

【0064】



図 8 にコマンドキューのブロックを示した。

【0065】

201 はコマンドを格納するレジスタ群で、CPU バスより入力された、`Cm d I D`、`read__not__write`、`burst__not__single`、`address`、`WriteData`、`Byte__enable`などが、格納される。202 はバリッドビットであり、情報が書き込まれる場合に 1 が書き込まれ、キューがシフトされると、0 が書き込まれる。203 は `Command__Q` がフルの状態を示す信号で、`cpubus__Cmd r d y p` に接続され、CPU (2001) に対し、バストランザクションの発行を停止させる。206 はキュー管理のシーケンサで、ライト要求 204、シフト要求 205 により、キューの管理を行う。207 は `Q__v a l i d` 信号であり、`Decode (102)` に接続される。

【0066】

`Decode (102)` は、アドレスデコーダであり、コマンドキュー (101) の出力を元にデコードを行う。デコード結果には、ソースが CPU0 か CPU1 か、ターゲットデバイスが、メモリか IO か、さらに IO アクセスの場合、`RCIF`、`BIF`、`EBIF`、`SRIF` のいずれかを示した情報が含まれる (108)。

【0067】

`Dispatch (103)` はデコーダ (102) の結果を元に、コマンドキュー (101) 内の次にサービスされるべきリクエストが `Ybus` マスタブロック (104 及び 105) に対し発行可能かを判定し、可能な場合には、適切な `Ybus` マスタブロックに起動要求を発行する。リードトランザクションの発行に関する制限を図 7 に表す。

【0068】

この制限事項を実施するための情報の保持を行っているのが、`Pending Q (403)` であり、判断を行っているのが、`Dispatch (103)` である。`Pending Q (403)` の構成を図 10 に、`Dispatch (103)` の状態遷移を図 9 に示した。`Dispatch (103)` において、CPU バ

スよりアクセス要求のない場合は、シーケンサはIDLE（301）の状態にある。

【0069】

CPUバスよりアクセス要求が、キューイングされると、後述する、Pending Q（403）から出力されるCPU0及びCPU1のリードリクエスト保留の状態（109）と、Decode（102）から出力されるCPUリクエスト情報（108）に基づきIOもしくは、MEMのYBusMasterを起動する。

【0070】

具体的には、図7にしたがって、CPU0よりIOへのアクセス要求があった場合、YBusMaster MEM（105）から出力されるcpu0__in__mem（109内）信号がアクティブでなかった場合に、シングル転送要求の場合は、遷移311によりYBusMaster IO（104）に対して、シングル転送要求を発行するステートIOSingle（313）へ制御が移動し、YBusMaster IO（104）へ起動を要求した後、遷移312でアイドルへ戻る。バースト転送の場合も同様に、305→307→306の遷移により、IOのバースト転送の起動を行う。また、IOのリード要求の場合は、図10に示したPending Q（403）内部のTargetCheck回路（509）により、異なったIOターゲットに対するリード要求がすでに保留になっていないかがチェックされ、DispatchOK信号（513）によって、異なったターゲットへのリード要求に対するデータが戻ってくるまで、実際のトランザクションは発行されない。

【0071】

CPU0からのアクセス要求がMEMに対するものだった場合は、CPU0の要求がすでにYBusMaster IO（104）内部で保留になっていないかをCPU0__in__io信号によって、検出し、保留ではない場合に、バーストかシングルかに応じて、309→308→310もしくは、304→302→303の系によりメモリアクセス要求がYBusMaster MEM（105）に対して発行される。YBusMaster IO（104）、YBusMaster

rMEM (105) はそれぞれ、I/Oデバイス及びメモリに対するアクセスを行うYbusマスタブロックである。この2つのブロックは、内部的には、まったく同一の構成をとる。YBusMasterIO (104) 及びYBusMasterMEM (105) について、YBusMasterIO (104) を例にとって説明する。

【0072】

図11はYBusMasterIO (104) の内部構造を表す。Info Latch (401) は、Dispatch (103) からの起動要求によりYbusトランザクションを発行するのに必要な情報を必要期間保持する。これらの情報には、アドレス、バイトイネーブル、ライトデータ、リード、ライト、バーストトランザクション、シングルトランザクション、リクエスト発行CPUの番号等が含まれる。

【0073】

Info Latch (401) は同時に、情報のデコードを行い、YMasterSM (402) に対して、起動要求を発行する。

【0074】

YMasterSM (402) には、Ybusマスタステートマシンが内蔵され、Ybusに対して、シングルリード、シングルライト、バーストリード、バーストライトのいずれかのトランザクション要求を発行する。

【0075】

発行要求がリードであった場合には、Ybusに対してリードトランザクションを発行すると同時に、PendingQ (403) に対して、リードリクエストに使用した情報の保持を指令する。Ybusに発行したリクエストがライトの場合は、ライトデータをアドレス等のコントロール情報と同時にYbusに対してドライブし、シングルライトの場合は、転送を終了し、アイドル状態に戻る。

【0076】

バーストライトの場合は、アクセスタゲットデバイスよりレディ信号が返って来た次のクロックサイクルで、2ビット目のデータをドライブし、アイドル状態に戻る。

【0077】

Ybusより戻ってきたリードデータは、ReadReturnbgi c (404) 内のリードバッファに一時格納される。

【0078】

その後リターンデータをCPU (2001) に転送するため、ReadReturnArbitor (106) に対して、CPUバスのリードリターントランザクション発行要求が行われる。Read Return Arbitor (106) は、リードリターン要求のアービトレーションを行い、ReadDataMux (107) に対して、リードリターンシーケンスの開始と、データのセレクト信号を発行する。ReadDataMuxは、リードデータをCPUバス上にドライブすると同時に、cpubus__rdrdyp信号でCPUに対して、ReadDataの返送を通知する。

【0079】

【発明の効果】

以上説明したように、本発明によれば、保持された、命令発行CPUの番号、転送先、及び転送保留中のCPU番号に基づき、リード時に、リターンデータが転送命令の発行順と入れ替わる事を阻止し、一旦、共通バスを介し、シリアルライズされた転送を再度、複数の接続経路を用い同時並行に発行することができる。これにより、更に以下の効果を奏する。

【0080】

①バススヌーピングによるキャッシュのコヒーレンシ管理、アトミックトランザクションを性能の低下なしに実現する。

【0081】

②コマンドキュー（ライトバッファ）の実装を可能にし、CPUをバストランザクションから早期に開放することにより、性能の向上をもたらす。

【0082】

③一旦シリアルライズされたCPUのトランザクションを再度並列化することにより、低速のデバイスにアクセスしている一方のCPUの転送終了を待たずに、他方のCPUの転送を終了することが出来、性能低下を抑えることが出来る。

【 0 0 8 3 】

④ C P U バススレーブ部分の回路を共通化でき、複数の C P U バスを独立して持つ場合より、回路規模を小さく出来る。

といった効果が得られる。

【図面の簡単な説明】**【図 1】**

実施形態のシステムコントローラの全体構成を示すブロック図である。

【図 2】

本システムの実使用環境を表す図である。

【図 3】

バスブリッジの構成を示す図である。

【図 4】

システムバスブリッジの構成を表す図である。

【図 5】

C P U バスインターフェース部を表すブロック図である。

【図 6】

バスの転送の定義を示す図である。

【図 7】

リードトランザクションにおける発行の制限を示す図である。

【図 8】

コマンドキューの構成を表す図である。

【図 9】

デイバッチステートマシンの状態遷移を示す図である。

【図 1 0】

Y b u s リードペンディングキューを示した図である。

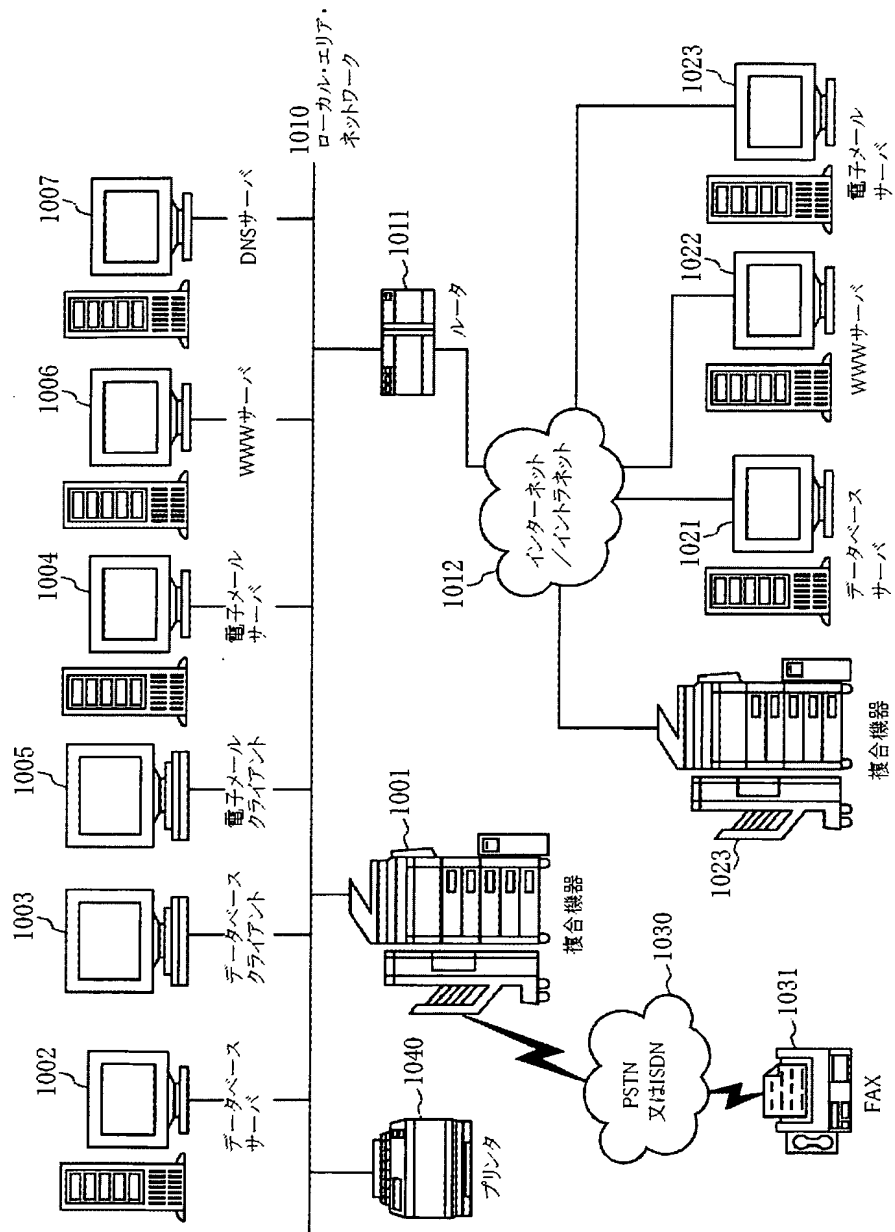
【図 1 1】

Y b u s マスタブロックのブロック図である。

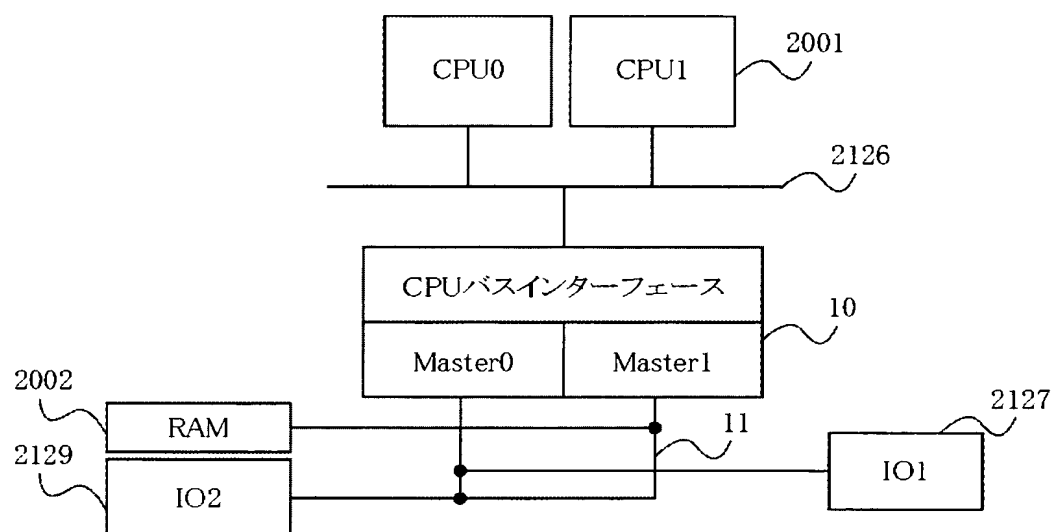
【図 1 2】

従来例の構成を示す図である。

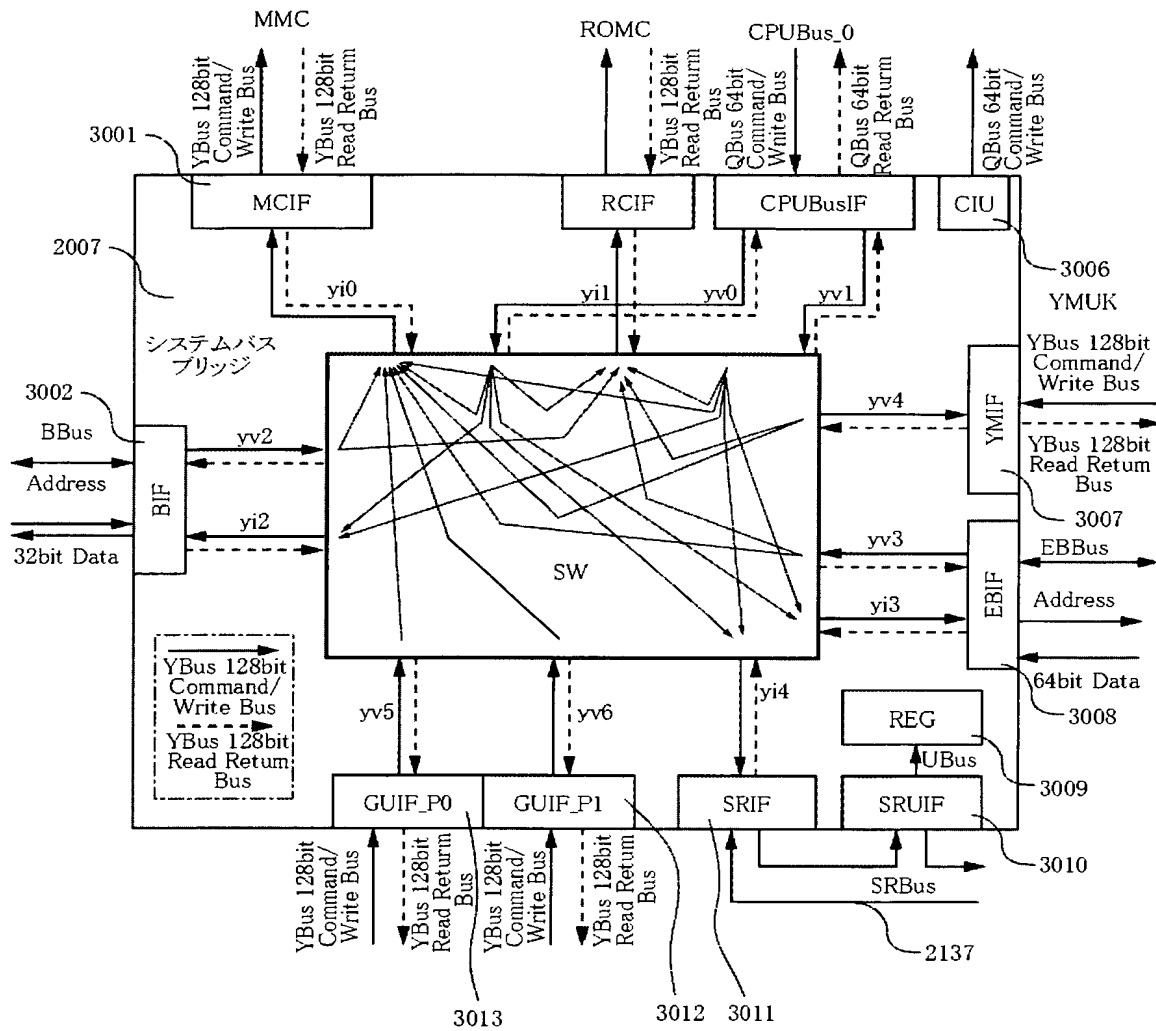
【図 2】



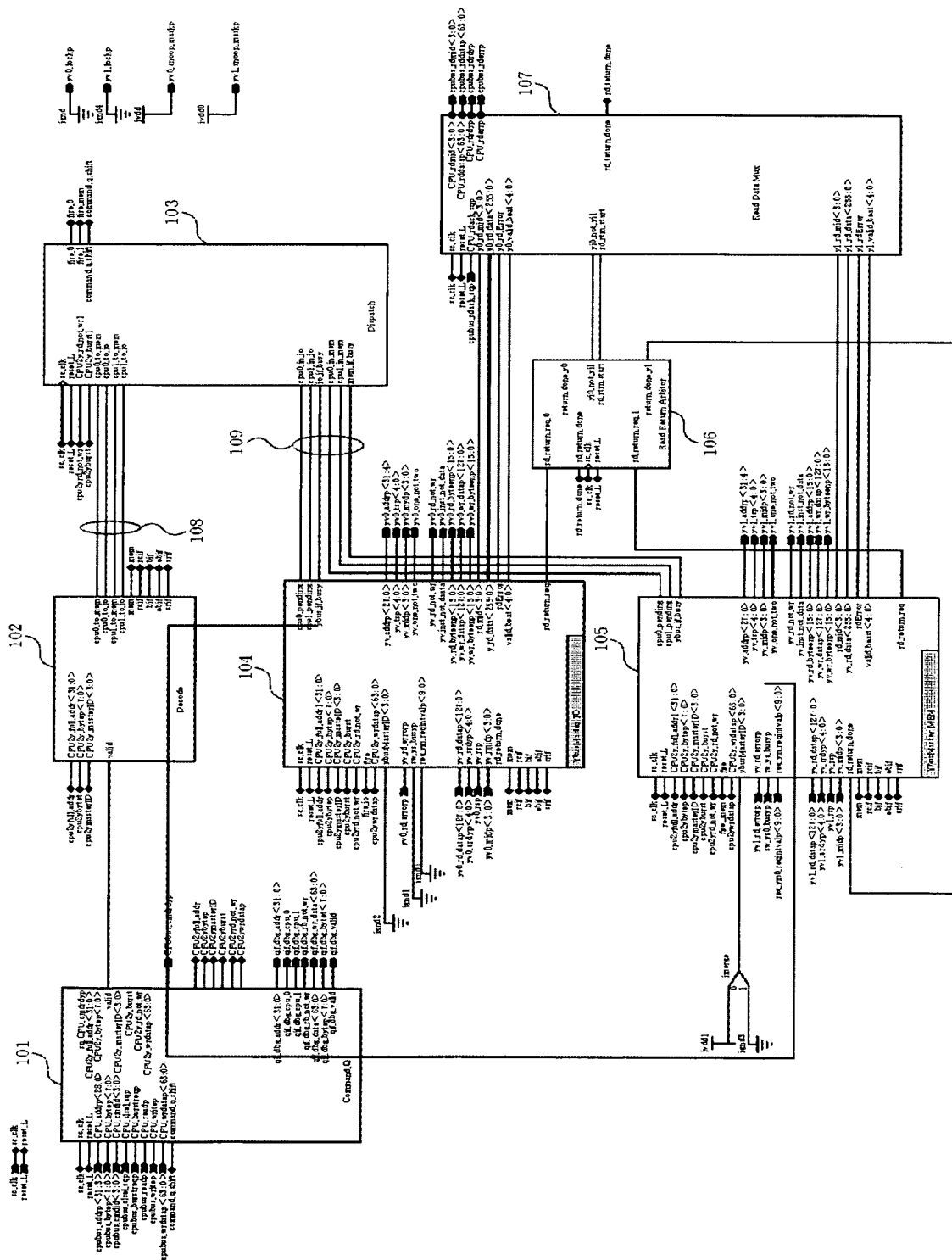
【図 3】



【図 4】



【図 5】



【図 6】

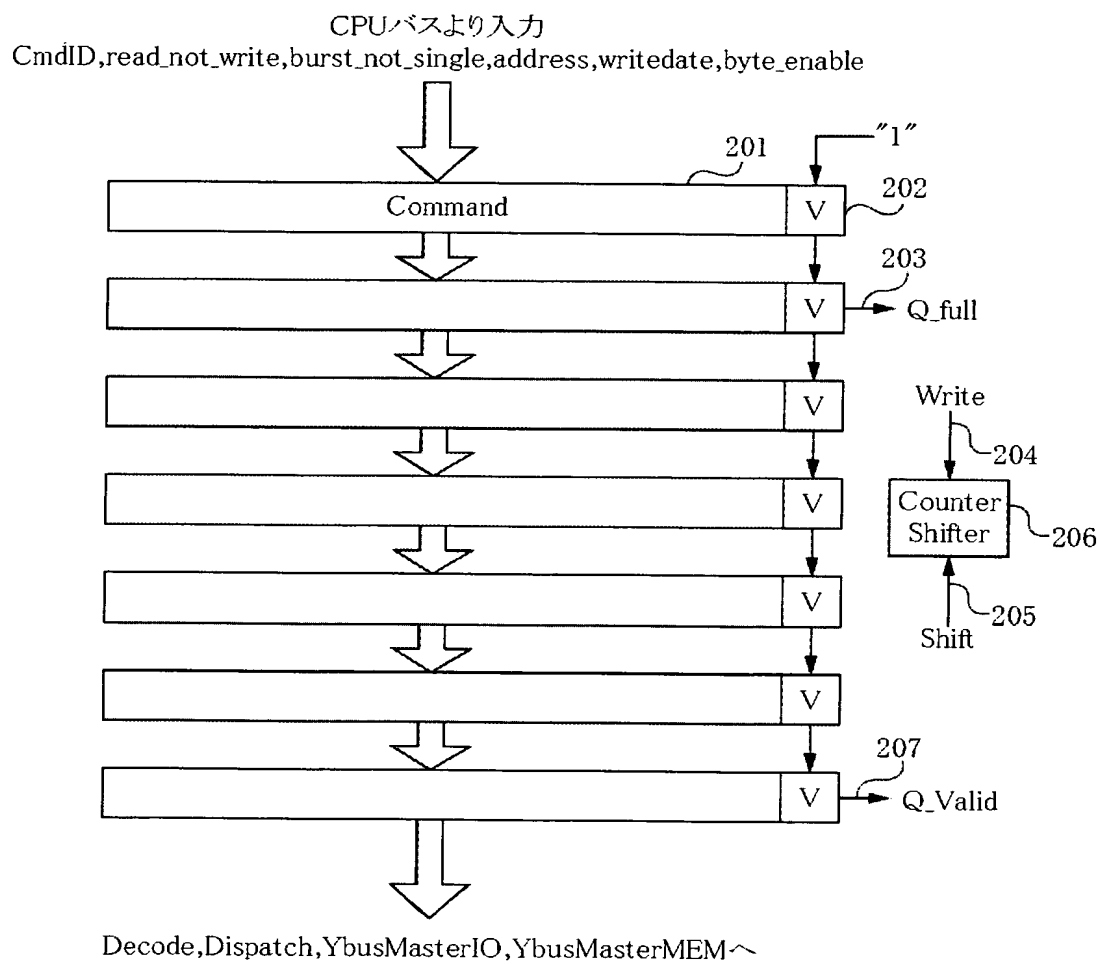
Transaction Type	Description	
Non-cached Instruction Read	Single read	
cached Instruction Read	4 beat burst read	Critical word first(CWF)
Non-cached Load Data Read	Single read	
cached Load Data Read	4 beat burst read	CWF
Store Data Write	Single write	
Store Data Write Burst	4 beat burst write	CWF



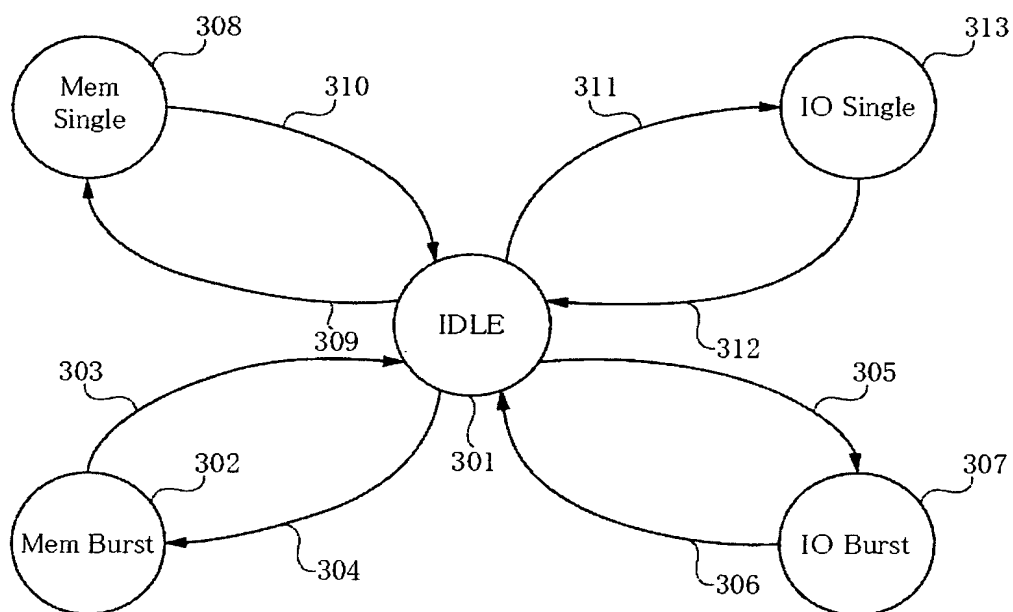
【図 7】

前回ソース	前回ターゲット	今回ソース	今回ターゲット	発行可否
CPU0	IO A	CPU0	IO A	○
CPU0	IO A	CPU0	IO A以外	×
CPU0	IO A	CPU0	MEM	×
CPU0	IO A	CPU1	IO A	○
CPU0	IO A	CPU1	IO A以外	×
CPU0	IO A	CPU1	MEM	○
CPU0	MEM	CPU0	IO	×
CPU0	MEM	CPU0	MEM	○
CPU0	MEM	CPU1	IO	○
CPU0	MEM	CPU1	MEM	○
CPU1(逆も同様)				

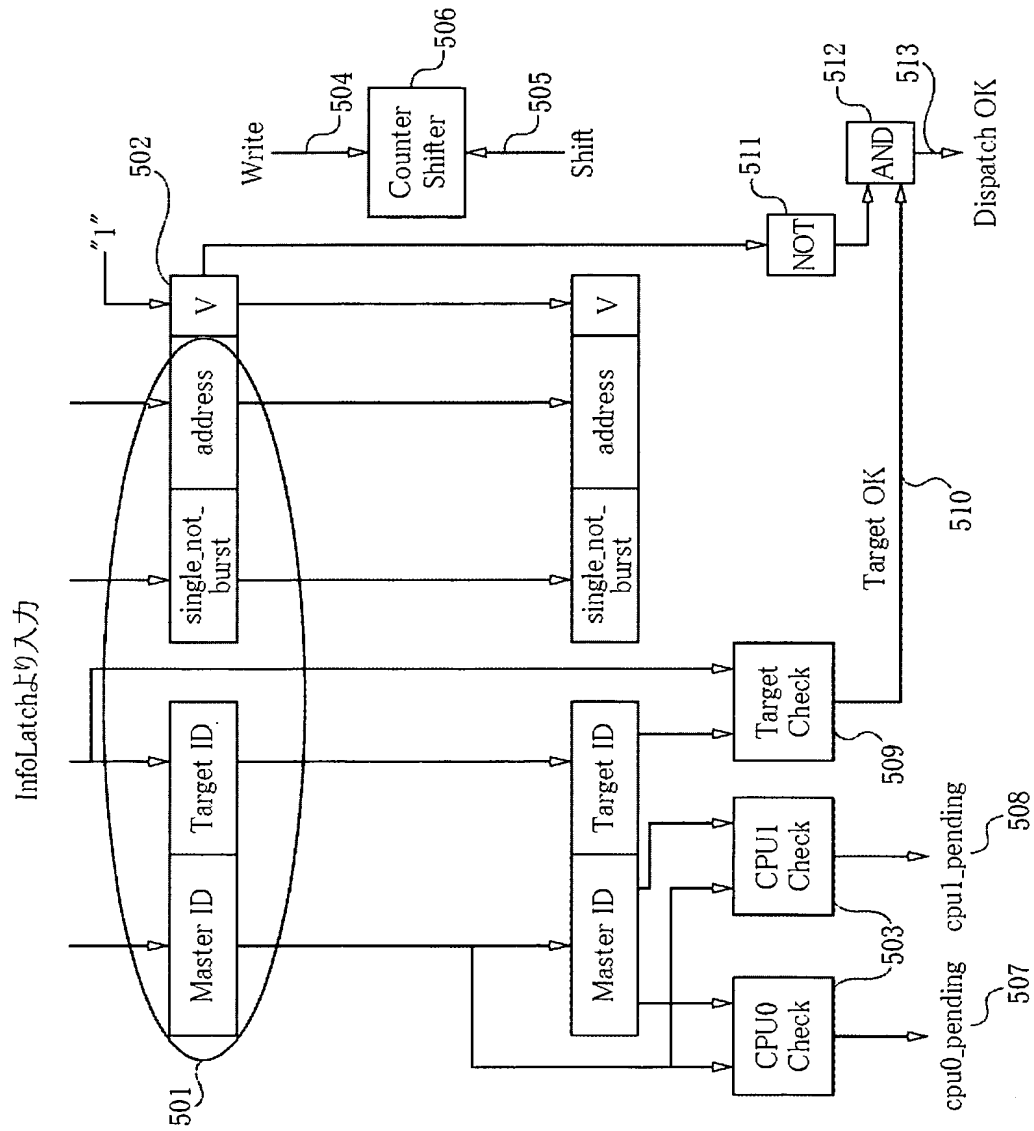
【図 8】



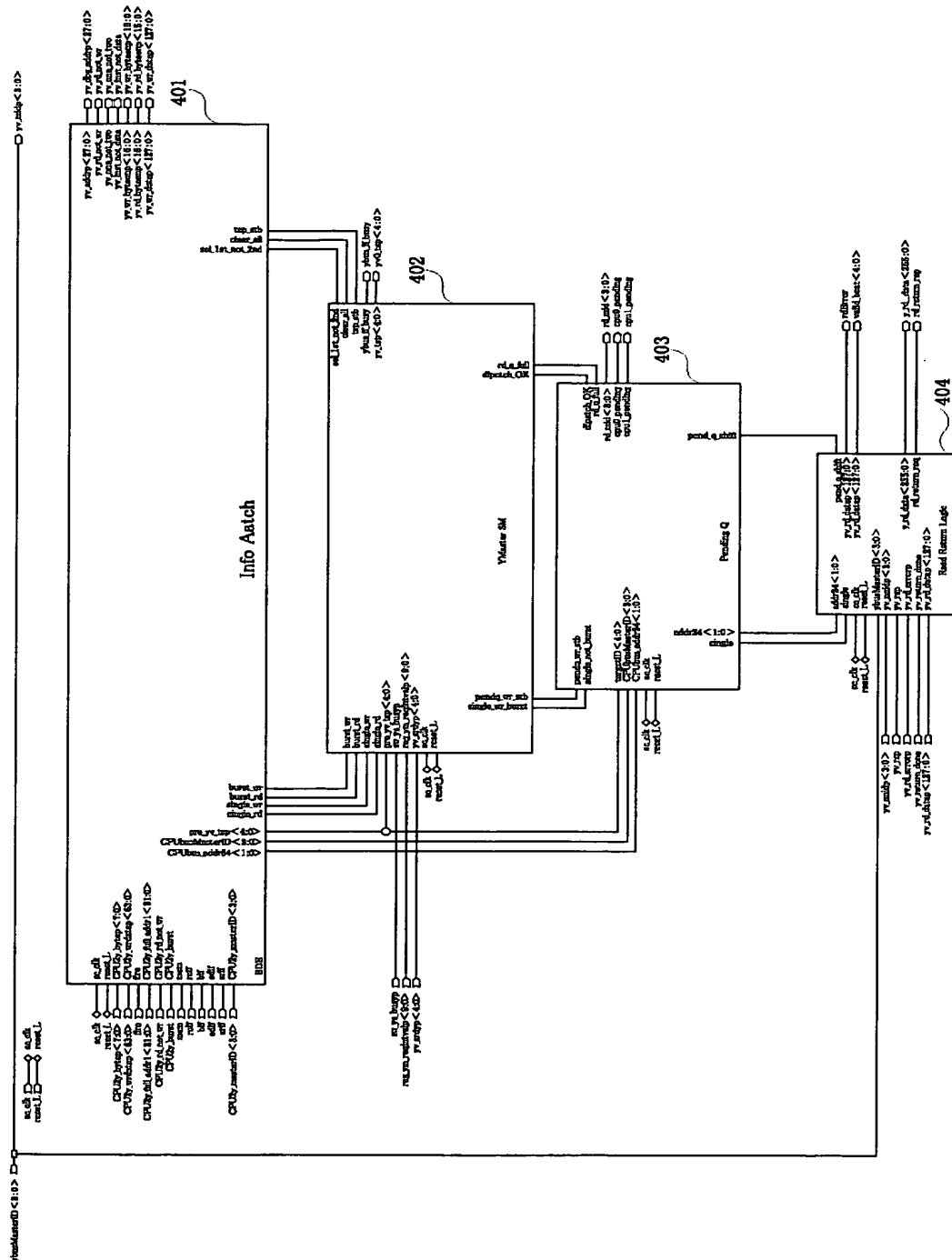
【図 9】



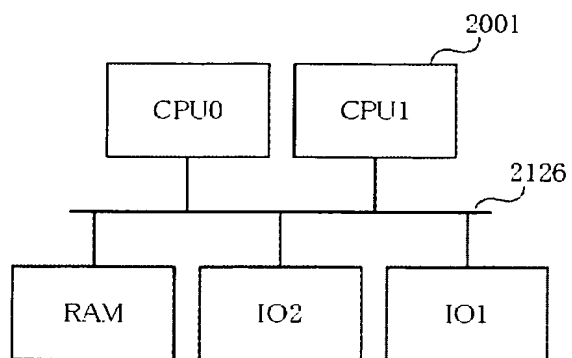
【図 10】



【図 1 1】



【図 1 2】



【書類名】 要約書

【要約】

【課題】 複数CPUを用いたシステムコントローラの性能を向上させる。

【解決手段】 共有バス2126を介して接続される複数のCPU2001と複数のメモリ2002またはIOデバイス2127、2129とを、リード命令とリードデータリターンとが分離可能なバスを介して接続されたシステムコントローラに、命令を発行したCPUと、その命令の転送先と、転送保留中のCPUとを識別する識別情報を保持する保持手段と、リード時に、前記保持手段の保持内容に基づいて、リターンデータが転送命令の発行順と入れ替わらないように制御する順序制御手段と、一旦シリアルライズされて前記共有バスを介して行なわれた転送を、複数の接続経路を用いて並行に発行する発行手段とを備える。

【選択図】 図3

特願 2 0 0 2 - 2 8 6 0 5 0

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 1 0 0 7]

1. 変更年月日

1 9 9 0 年 8 月 3 0 日

[変更理由]

新規登録

住 所

東京都大田区下丸子 3 丁目 3 0 番 2 号

氏 名

キャノン株式会社